Argument Outline – FPGA Paper

1. Introduction
   1. Purpose of paper is to explore best practices for designing a Sub-VT FPGA
      1. Why is a sub-VT FPGA worthwhile
         1. Smaller front-end cost (time and money)
   2. What practices we’re looking at
      1. Switch Choice
      2. Sense Amp Choice
      3. Sizing
      4. Intra-CLB Interconnect
   3. Also beginning to explore Dynamic Voltage Scaling (DVS)
      1. Why DVS would be beneficial
         1. Taking advantage of critical paths, no matter what the circuit design is (timing slack)
         2. Provide the lowest power approach to meet a timing constraint or a fastest approach to meet a power constraint
         3. Chip remains useful if constraints change
         4. Chip can function in multiple modes, each with different constraints
      2. DVS will include both VDDc (Configuration Bit) voltage as well as global VDD
      3. To begin the DVS process, we need to find what voltages we will DVS to
2. Switch Choice
   1. Choices
      1. Purely PG interconnect
      2. Purely TX interconnect
      3. Purely Buffer interconnect
   2. Metrics of Interest
      1. Power
      2. Delay
      3. Area
      4. Robustness (susceptibility to variation)
   3. PG interconnect (in Sub-VT) minimizes delay and energy
      1. Problem – doesn’t work for some complicated scenarios
      2. Possible Solutions
         1. Include repeaters in the fabric (optimization problem to see what choice minimizes energy, delay, and area)
         2. Upsize the drivers (optimization between delay, energy, and area again)
         3. Combination of the two
3. Sense Amp
   1. Choices
      1. Modified Schmitt Trigger
      2. MST w/o Hysteresis
      3. Custom SA1 – (muxed-SA)
      4. Custom SA2 – (Aatmesh)
   2. Metrics of interest
      1. Power
      2. Delay
      3. Area
      4. Robustness (susceptibility to variation)
   3. Currently, Custom SA1 minimizes energy and delay (ED Curve)
      1. Haven’t gotten Aatmesh’s working in simulation
4. Sizing
   1. Former literature did sizing for FPGAs, but not necessarily at the Sub-VT range
      1. Using the same sizing as nominal, large (0.35 um) processes is not sufficient in smaller technologies at the Sub-VT regime
      2. Should be sized to minimize delay without hurting energy and area too much
      3. Important things to be sized
         1. I/O Drivers
         2. CLB Drivers
         3. Switches
         4. Connection Boxes
      4. Possible methods for sizing optimizations
         1. Minimize product of delay, energy, and area
         2. Do ED Curves (sweeping size), but multiply each curve by the area of the device
5. Intra-CLB Interconnect
   1. It is shown that clustering provides increased efficiency for FPGAs by decreasing the amount of global routing necessary
   2. If intra-clb interconnect could be optimized, then either
      1. Greater degree of clustering to further minimize inefficient global interconnect
      2. Same amount of clustering, and higher efficiency in the clusters is higher efficiency overall for the FPGA
   3. Proposed structures
      1. mini-FPGA structure
         1. arranging the CLBs in rows and columns, like the FPGA
      2. 6-nearest-neighbor structure
   4. Proposed method for testing
      1. Create simulation
      2. Find total number of nets through simulation
      3. Find total length of nets by approximation from geometry (or from paper resources)
      4. Numbers will be equations relating to N (number of CLBs in the cluster)
   5. Hunch – mini-FPGA will be most efficient structure
6. Dynamic Voltage Scaling
   1. Developed Methodology/Tool Flow for finding optimal VDD/VDDc pairs for any global net in an FPGA
   2. This knowledge provides the destinations of the voltage scaling that we will soon do